

WHAT IS CLAIMED IS:

- 1                   1. An interface between a master and one or more slave modules comprising:  
2                   a master;  
3                   a slave having a set of addressable registers including egress mailbox registers,  
4 ingress mailbox registers, and indirect access address registers;  
5                   a direct memory access (DMA) engine coupled to the master by a first bus and  
6 to the slave by a second bus, with the second bus comprising:  
7                   a set of bi-directional data lines for transmitting data between the slave and the  
8 DMA engine;  
9                   a set of master address lines for transmitting address data from the DMA  
10 engine to the slave;  
11                  a master data strobe for strobing data;  
12                  a master read/write signal for indicating whether data is to be read from or  
13 written to the slave;  
14                  a set of slave select signals for selecting one of a plurality of slaves connected  
15 to the second bus;  
16                  a slave wait signal asserted by a slave to delay a data transfer;  
17                  a slave reset signal,  
18                  a clock output signal, and  
19                  a clock input signal;  
20                  where the DMA engine performs direct data transfers to the slave by asserting  
21 a slave select signal to the slave and transferring data over the set of bi-directional data lines  
22 to the slave egress or ingress data registers and performs indirect data transfers to slave  
23 memory by writing address data over the set of bi-directional data lines to the indirect address  
24 register of the slave and where the slave utilizes its own memory map and the address data to  
25 transfer data between a location indicated by the address data and the DMA engine.
- 1                   2. The interface of claim 1 where the DMA engine negotiates with a slave to  
2 implement either an asynchronous, synchronous, or source synchronous data transfer.
- 1                   3. The interface of claim 1 where the DMA engine negotiates with all slaves  
2 during reset to determine the maximum bus width available to transfer data.
4. The interface of claim 1 where:

the slave includes status register and message signal interrupt (MSI) register;  
and

where the slave asserts a bit in the status register to indicate it is ready for a transaction and where the DMA engine asserts a bit in the MSI register to indicate when a transaction is complete.

1                   5. A method for allowing a DMA engine to provide access to a plurality of  
2 slave devices to multiple masters, the protocol, implemented by hardware and software on the  
3 DMA engine, the master, and the slave devices, comprising the steps of:  
4                   to implement a direct message transfer to a slave device:  
5                   accessing a slave status register to read a direct message ready status bit which  
6 is set when the slave is ready to transfer data;  
7                   transferring message data using the DMA engine and a slave mailbox register  
8 if the direct message ready status bit is set;  
9                   setting an message transfer complete status interrupt at the slave to indicate  
10 when the transfer of the message is complete; and  
11                  to implement an indirect data transfer to the memory space of a slave device:  
12                  accessing a slave status register to read an indirect message ready status bit  
13 which is set when the slave is ready to transfer data;  
14                  transferring address data using the DMA engine and slave indirect address  
15 mailbox register if the indirect message ready status bit is set;  
16                  setting an indirect transfer message interrupt bit at the slave to initiate the  
17 indirect transfer;  
18                  transferring message data between the DMA engine and slave mailbox  
19 registers if the indirect message ready status bit is set, where the slave utilizes its own  
20 memory map and the address data to transfer data between a location indicated by the address  
21 data and the DMA engine; and  
22                  setting an message transfer complete status interrupt at the slave to indicate  
23 when the transfer of the message is complete.

1                   6. The method of claim 5 further comprising the step of:  
2                   negotiating with all the slaves to implement either an asynchronous,  
3 synchronous, or source synchronous data transfer.

1                   7. The method of claim 5 further comprising the step of:  
2                   starting the bus upon reset at a fixed bus-width and then negotiating with all  
3 the slaves to implement acceptable bus bit-width.

1                   8. A system for allowing a DMA engine to provide access to a plurality of  
2 slave devices to multiple masters, the protocol, implemented by hardware and software on the  
3 DMA engine, the master, and the slave devices, said system comprising:  
4                   means for implementing a direct message transfer to a slave device including:  
5                   means for accessing a slave status register to read a direct message ready  
6 status bit which is set when the slave is ready to transfer data;  
7                   means for transferring message data using the DMA engine and a slave  
8 mailbox register if the direct message ready status bit is set;  
9                   means for setting an message transfer complete status interrupt at the slave to  
10 indicate when the transfer of the message is complete; and  
11                  means for implement an indirect data transfer to the memory space of a slave  
12 device including:  
13                  means for accessing a slave status register to read an indirect message ready  
14 status bit which is set when the slave is ready to transfer data;  
15                  means for transferring address data using the DMA engine and slave indirect  
16 address mailbox register if the indirect message ready status bit is set;  
17                  means for setting an indirect transfer message interrupt bit at the slave to  
18 initiate the indirect transfer;  
19                  means for transferring message data between the DMA engine and slave  
20 mailbox registers if the indirect message ready status bit is set, where the slave utilizes its  
21 own memory map and the address data to transfer data between a location indicated by the  
22 address data and the DMA engine; and  
23                  means for setting an message transfer complete status interrupt at the slave to  
24 indicate when the transfer of the message is complete.

1                   9. The system of claim 8 further comprising:  
2                   means for negotiating with all the slaves to implement either an asynchronous,  
3 synchronous, or source synchronous data transfer.

1                   10. The system of claim 8 further comprising:

- 2 means for starting the bus upon reset at a fixed bus-width and then negotiating
- 3 with all the slaves to implement acceptable bus bit-width.